

Nama : Riska Najla Nabilah
Kelas : 2IA11
NPM : 51422457
Mata Kuliah : Pemrograman Berbasis Objek

RANGKUMAN MATERI PERTEMUAN 1-6

A. Pengenalan Pemrograman Berbasis Objek

Pemrograman berorientasi objek (OOP) adalah paradigma pemrograman yang menekankan pembuatan program melalui objek, entitas yang memiliki atribut, karakter, dan kadang disertai kondisi. OOP berkembang dari pemrograman terstruktur pada tahun 1960 yang dikembangkan dari bahasa C dan Pascal. Dengan program terstruktur inilah mampu menulis program yang begitu sulit dengan lebih mudah. Pemrograman berorientasi objek tidak fokus pada cara terstruktur dalam menyelesaikan masalah, tetapi pada objek-objek yang dapat menyelesaikan masalah tersebut. Pemrograman berorientasi objek bekerja dengan baik ketika dibarengi dengan Objek-Oriented Analysis And Design Process (OOAD).

▪ Perbedaan Pemrograman Berbasis Objek dan Pemrograman Terstruktur

Pemrograman Berbasis Objek :

Menggabungkan fungsi dan data dalam kelas atau objek-objek dengan memiliki ciri **Encapsulation** (pengemasan), **Inheritance** (penurunan sifat), dan **Polymorphism** (perbedaan bentuk dan perilaku). Struktur program menjadi lebih ringkas dengan membuat object dan class, lalu bekerja berdasarkan object dan class tersebut. Kode program menjadi sangat reusable karena object dan class dapat digunakan berkali-kali, sehingga menghemat ruang memori.

Pemrograman Terstruktur :

Memecah program dalam fungsi dan data dengan memiliki ciri **Sequence** (berurutan), **Selection** (pemilihan), dan **Repetition** (perulangan). Struktur program rumit karena berupa urutan proses dan fungsi-fungsi. Re-use kode program cenderung kurang optimal.

▪ Class pada PBO

Class adalah cetak biru atau blueprint dari object. Class digunakan hanya untuk membuat kerangka dasar. Misal, class diibaratkan sebagai kendaraan. Kita tahu bahwa kendaraan memiliki ciri-ciri seperti merk, memiliki mesin, memiliki ban, dan beberapa ciri khas lain yang menyatakan sebuah benda tersebut adalah kendaraan. Selain memiliki ciri-ciri, sebuah kendaraan juga bisa dikenakan tindakan, seperti: menghidupkan mesin atau mematikan mesin.

Contoh Program :

```
// Deklarasi Class
public class Kendaraan{
    // Ini Adalah Class, semua konstruktor, variabel, method berada disini
}
```

- Objek dalam Pemrograman

Objek mirip dengan suatu rekaman (record) dalam suatu sistem berkas. Objek adalah entitas yang memiliki atribut, karakter dan kadang kala disertai kondisi. Objek mempresentasikan sesuai kenyataan seperti siswa, mempresentasikan dalam bentuk konsep seperti merek dagang, juga bisa menyatakan visualisasi seperti bentuk huruf (font).

Dalam pengembangan perangkat lunak berorientasi objek, objek dalam perangkat lunak akan menyimpan state-nya dalam variable dan menyimpan informasi **tingkah laku (behaviour) dalam method atau fungsi-fungsi/prosedur.**

- Objek pada PBO

Untuk membuat objek, bisa menggunakan kata kunci new yang digunakan untuk membuat objek baru, selanjutnya menentukan 3 langkah untuk membuat sebuah objek. Yaitu: mendeklarasikan variable, membuat objek baru (Instansiasi) dan memanggil konstruktor.

Contoh Program :

```
public class Kendaraan{  
    // Konstruktor Dengan Parameter  
    public Kendaraan(String nama){  
        System.out.println("Nama Kendaraannya Adalah "+ nama);  
    }  
  
    public static void main(String[] args){  
        // Perintah untuk membuat objek jenis  
        Kendaraan jenis = new Kendaraan("Pesawat Terbang");  
    }  
}  
  
// Output = Nama Kendaraannya Adalah Pesawat Terbang
```

B. Inheritance

Pewarisan merupakan sebuah bentuk “penggunaan kembali” (reusability); dimana class baru dibuat dari class yang pernah ada yang (biasanya) ditambah fasilitasnya.

- Pewarisan tunggal (single inheritance) merupakan pewarisan dari satu class pokok (induk).
- Pewarisan ganda (multiple inheritance) merupakan pewarisan dari dua atau lebih class pokok.

Class yang mewariskan disebut dengan **superclass/parent class /base class**, sedangkan class yang mewarisi (class yang baru) disebut dengan **subclass/child class/derived class**. Untuk menerapkan inheritance, gunakan statement “**extends**”. Keyword “**super**” digunakan oleh subclass untuk memanggil constructor, atribut dan method yang ada pada superclass-nya.

Contoh untuk memanggil constructor milik superclass-nya :

```
super()  
super(parameter)
```

Contoh untuk memanggil atribut dan method milik superclass-nya :

```
super.namaAtribut  
super.namaMethod(parameter)
```

C. Encapsulation

Enkapsulasi (encapsulation) adalah cara untuk melindungi property (atribut) / metode tertentu dari sebuah kelas agar tidak dapat diakses atau dimodifikasi secara sembarangan oleh bagian program lainnya. Cara untuk melindungi data yaitu dengan menggunakan access modifiers (hak akses). Ada 4 hak akses yang tersedia, yaitu :

No	Modifier	Pada class dan interface	Pada method dan variabel
1	Default (tidak ada modifier)	Dapat diakses oleh yang sepaket	Diwarisi oleh subkelas dipaket yang sama, dapat diakses oleh method-method yang sepaket
2	Public	Dapat diakses dimanapun	Diwarisi oleh subkelasnya, dapat diakses dimanapun
3	Protected	Tidak bisa diterapkan	Diwarisi oleh subkelasnya, dapat diakses oleh method-method yang sepaket
4	private	Tidak bisa diterapkan	Tidak dapat diakses dimanapun kecuali oleh method-method yang ada dalam kelas itu sendiri

Aksesabilitas	public	private	protected	default
Dari kelas yang sama	Ya	Ya	Ya	Ya
Dari sembarang kelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari sembarang kelas di luar paket	Ya	Tidak	Tidak	Tidak
Dari subkelas dalam paket yang sama	Ya	Tidak	Ya	Ya
Dari subkelas di luar paket	Ya	Tidak	Ya	Tidak

D. Polymorphism

➤ Overloading

Diperbolehkannya dalam sebuah class memiliki lebih dari satu nama function/method yang sama tetapi memiliki parameter/argument yang berbeda.

```
1 public class Overloading {
2     public void Tampil(){
3         System.out.println("I love Java");
4     }
5     public void Tampil(int i){
6         System.out.println("Method dengan 1 parameter = "+i);
7     }
8     public void Tampil(int i, int j){
9         System.out.println("Method dengan 2 parameter = "+i+ " & "+j);
10    }
11    public void Tampil(String str){
12        System.out.println(str);
13    }
14
15    public static void main(String a[]){
16        Overloading objek = new Overloading();
17        objek.Tampil();
18        objek.Tampil(8);
19        objek.Tampil(6,7);
20        objek.Tampil("Hello world");
21    }
22 }
```

➤ Overriding

Kemampuan dari subclass untuk memodifikasi method dari superclass-nya, yaitu dengan cara menumpuk (mendefinisikan kembali) method superclass-nya. Contoh overriding method dapat dilihat pada subclass “Mobil” yang mendefinisikan kembali method keterangan() dan hapus() dari class “Kendaraan”.